RETHINKING GFLOWNET TRAINING: NEGATIVE REINFORCE DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative Flow Networks (GFlowNets) are models that learn to sample discrete objects proportional to a target reward. They are trained by minimizing balance errors over sampled trajectories, which inherently ties correctness to exploration: if a region is not visited, balance cannot be enforced there. Adaptive exploration is therefore crucial, yet standard training relies on a semi-gradient update that treats the sampling distribution as fixed at each optimization step. On the other hand, the full gradient reveals a second mechanism that reduces the observed errors in an anti-exploratory manner. In this paper, we establish a connection between the full gradient and min-min dynamics between the GFlowNet and a decoupled sampler, showing that the sampler reduces observed violations by avoiding high-error regions. This insight motivates a bridge between two-player games and single-policy dynamics, which underpins the design of our proposed method, Negative Reinforce Dynamics (NRD). By tying the parameters of an adversarial game together, we recover a single-policy dynamic that extends exploration while maintaining correctness. In standard benchmarks, our method outperforms existing approaches in both convergence speed and exploration efficiency, without training a separate teacher or auxiliary exploration policy.

1 INTRODUCTION

Generative Flow Networks (GFlowNets) learn stochastic policies that construct structured objects with probability proportional to a nonnegative reward (Bengio et al., 2021; 2023). Their ability to generate diverse high-reward samples has made them useful in applications ranging from biological sequence design (Jain et al., 2022) and Bayesian structure learning (Deleu et al., 2023) to recent approaches for language-model post-training (Hu et al., 2024; Bartoldson et al., 2025).

Training a GFlowNet to sample from the target distribution is formulated by enforcing balance conditions, with correctness holding if and only if balance is satisfied globally (Bengio et al., 2023). During training, however, balance can only be corrected on trajectories that are observed, leaving poorly visited regions potentially uncorrected (Madan et al., 2025; Dall'Antonia et al., 2026b). To mitigate this problem, training trajectories are often collected from an exploratory sampler, such as an ε-greedy policy, in order to broaden coverage (Whitammer et al., 2022). Previous work has furthermore introduced exploration mechanisms that use auxiliary samplers or teacher policies to expose the GFlowNet to regions where balance errors remain large (Kim et al., 2025; Dall'Antonia et al., 2026a).

In this paper we opt for a different path, revisiting how we solve the balance conditions in GFlowNet training. In the usual training pipeline, balance is enforced by turning local violations into a loss and minimizing its expectation over a sampling policy. However, with an adaptive sampling distribution that evolves during training, decreasing this expected loss may not correspond to correcting the underlying violations: if the policy adapts to avoid high-loss regions, then we will reduce the expected loss while retaining significant balance violations. This can be seen by analyzing the gradient of this expected loss, when differentiating through a parameter-dependent sampler Q_θ, using the score-function identity (Williams, 1992):

$$\nabla_\theta \mathbb{E}_{\tau\sim Q_\theta}[L_\theta(\tau)] = \underbrace{\mathbb{E}_{Q_\theta}[\nabla_\theta L_\theta(\tau)]}_{\text{semi-gradient}} + \underbrace{\mathbb{E}_{Q_\theta}[L_\theta(\tau)\nabla_\theta \log Q_\theta(\tau)]}_{\text{score term}}. \tag{1}$$

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

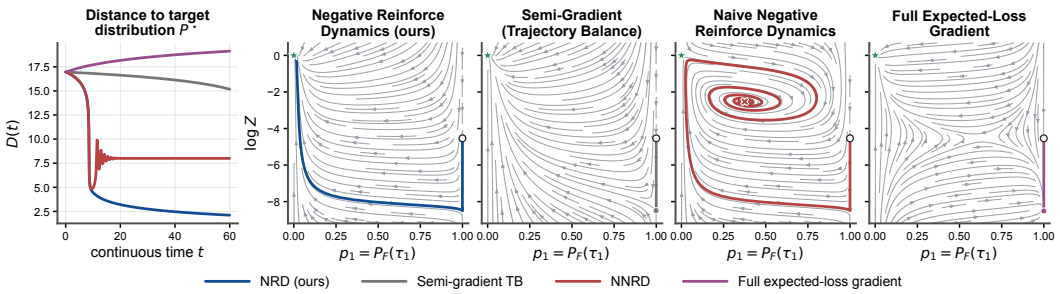

Figure 1: **Two-trajectory dynamics.** Left: log-ratio discrepancy over continuous time. Right: vector fields and trajectories in $(p_1, \log Z)$ from a common initialization. NRD reduces the discrepancy faster than semi-gradient TB in this example, while Naive NRD (NNRD) and the full expected-loss gradient fail to reach the balanced solution (green star).

This reveals two mechanisms for decreasing the loss: one which corrects balance violations (the semi-gradient term) and one which avoids high-loss regions (the score term). Standard GFlowNet training keeps only the former, by treating the sampling distribution as fixed during differentiation; this yields the usual semi-gradient update. The second term reveals an additional mechanism through which the sampling distribution can reduce the expected loss, which, under ordinary minimization, lowers the objective without enforcing balance – which is generally undesirable.

Nevertheless, we ask whether this mechanism can instead be turned into a beneficial exploration signal. To do so, we reframe the full-gradient dynamics encapsulated in Equation 1 as a 'weight-tied' version of a cooperative two-player game between the GFlowNet and the sampling policy, and argue that it is this cooperation that leads to the harmful score term. To rectify this we formulate a second game, this time *adversarial*, which yields a novel *beneficial* exploration mechanism. By appropriately tying the two players' weights we obtain our proposed scheme, *Negative Reinforce Dynamics (NRD)*, which we find to perform significantly better than standard trajectory balance training, at little additional cost.

We establish theoretical guarantees for these dynamics in tractable population models. In the free-trajectory setting, NRD converges globally to the balanced solution and admits a pointwise log-ratio discrepancy comparison with semi-gradient Trajectory Balance. In one-dimensional exponential families under fast normalization, NRD reaches any positive Jeffreys-divergence threshold sooner from a common initialization, with an unbounded additive advantage in the two-trajectory setting.

Finally, we evaluate NRD across multimodal distribution-matching and structured-generation tasks. The method improves convergence to target distributions and finite-budget exploration relative to standard Trajectory Balance and comparable-cost baselines, while remaining competitive with approaches that train auxiliary exploration policies. Despite its two-player motivation, NRD requires only a sign-selective score correction on top of the standard Trajectory Balance update and does not require a separately trained sampler.

2 Background

Let $\mathcal{X}$ be a finite set of terminal objects and $R(x) > 0$ an unnormalized reward. A GFlowNet aims to sample from the target terminal distribution $P^\star(x) = R(x)/Z^\star$ (Bengio et al., 2021; 2023). To do so, we train a forward policy $P_F(\cdot \mid s)$ to construct a terminal object through a trajectory of intermediate states $\tau = (s_0, s_1, \ldots, s_T = x)$ on a directed acyclic state graph. The policy assigns probabilities to successive child states, inducing distributions over trajectories and terminal objects, denoted $P_F(\tau)$ and $P_T(x)$, respectively.

GFlowNet training is characterized by enforcing balance conditions that hold across all trajectories. Trajectory Balance (Whittaker et al., 2022) is one such condition, which can be expressed in terms of the forward policy, the reward, a learned backward policy P_B and a scalar Z:

$$ZP_F(\tau) = R(x)P_B(\tau \mid x), \quad \forall \tau \to x \in \mathcal{X}. \tag{2}$$

If this holds true then summing over all trajectories ending at a fixed object x gives $P_T(x) = R(x)/Z^\star$; the marginalization and normalization argument is given in Appendix D.1.

However, the balance condition alone does not provide a training objective. To achieve this, we convert it into a regression problem through a log-ratio residual and its squared loss (Whitammer et al., 2022):

$$\delta_\theta(\tau) := \log \frac{Z_\theta P_F^\theta(\tau)}{R(x) P_B^\theta(\tau \mid x)}, \qquad L_\theta(\tau) := \delta_\theta(\tau)^2. \tag{3}$$

The residual vanishes exactly when Trajectory Balance holds on τ, while squaring produces a nonnegative loss. Because the number of complete trajectories can be combinatorially large, the loss cannot generally be evaluated everywhere. Instead, trajectories are drawn from a sampling distribution Q, defining the expected balance loss

$$\mathcal{L}_Q(\theta) := \mathbb{E}_{\tau \sim Q}[L_\theta(\tau)]. \tag{4}$$

If Q has full support over trajectories, nonnegativity implies that $\mathcal{L}_Q(\theta) = 0$ if and only if $L_\theta(\tau) = 0$ for all τ. Thus, every full-support sampler identifies the same globally balanced solutions (Madan et al., 2023). For a fixed training distribution Q, the gradient of the expected Trajectory Balance loss is

$$\nabla_\theta \mathcal{L}_Q(\theta) = \mathbb{E}_{\tau \sim Q}[\nabla_\theta L_\theta(\tau)]. \tag{5}$$

Standard GFlowNet training is not generally performed with a fixed Q, but rather with an adaptive sampler that depends on the forward policy Kim et al. (2025); Dall'Antonia et al. (2026a). Nevertheless, each update treats the current sampling distribution as fixed, retaining the semi-gradient as above.

3 Rethinking GFlowNet Training

In this section, we investigate how the score mechanism can be used to shape exploration. We first separate the GFlowNet from its sampler to make their distinct roles explicit, and then study what happens when their dynamics are tied back into a single policy. This reveals a correspondence between GFlowNet–sampler games and single-policy updates that will motivate our method in the next section.

3.1 Expected Loss as a Surrogate

GFlowNet correctness requires satisfying balance conditions across all trajectories leading to a terminal object, a requirement that practical training translates into minimizing an expected loss under a full-support sampling policy Q_θ. While the expected loss serves as a natural surrogate, we show that there are two distinct mechanisms for minimizing it, one of which is fundamentally misaligned with the true goal of sampling from the target distribution. The score-function identity (Williams, 1992) decomposes the full gradient into two components, as derived in Appendix D.2:

$$\nabla_\theta \mathbb{E}_{\tau \sim Q_\theta}[L_\theta(\tau)] = \underbrace{\mathbb{E}_{Q_\theta}[\nabla_\theta L_\theta(\tau)]}_{\text{semi-gradient}} + \underbrace{\mathbb{E}_{Q_\theta}[L_\theta(\tau) \nabla_\theta \log Q_\theta(\tau)]}_{\text{score term}}. \tag{6}$$

Standard training uses only the semi-gradient term, which corresponds to treating the sampling distribution Q_θ as fixed during gradient computation. This restricts the optimization to the benign mechanism which focuses on correcting the balance errors along the observed trajectories. In the strictly on-policy setting, the semi-gradient can be seen as variational inference (Whitammer et al., 2023):

$$\mathbb{E}_{\tau \sim P_F^\theta}[\nabla_{\theta_F} L_\theta(\tau)] = 2 \nabla_{\theta_F} D_{\mathrm{KL}} \left(P_F^\theta(\tau) \,\middle\|\, \frac{R(x)}{Z^\star} P_B^\theta(\tau|x) \right), \tag{7}$$

where the derivative with respect to the forward parameters holds the backward policy and Z fixed. Appendix D.3 proves the identity, which interprets the standard update as steering the forward policy toward the target distribution. The score term, on the other hand, reduces the expected loss by changing which trajectories are sampled. Under ordinary minimization, it shifts probability toward regions with smaller balance error, allowing the objective to decrease without directly correcting the violations.

Although the full-gradient update is misaligned with the goal of correcting balance globally, its additional score term need not be viewed only as a failure mode: it provides a direct mechanism through which balance errors can shape the sampling policy. The question is therefore whether this mechanism can be used in a direction that supports, rather than opposes, global balance correction.

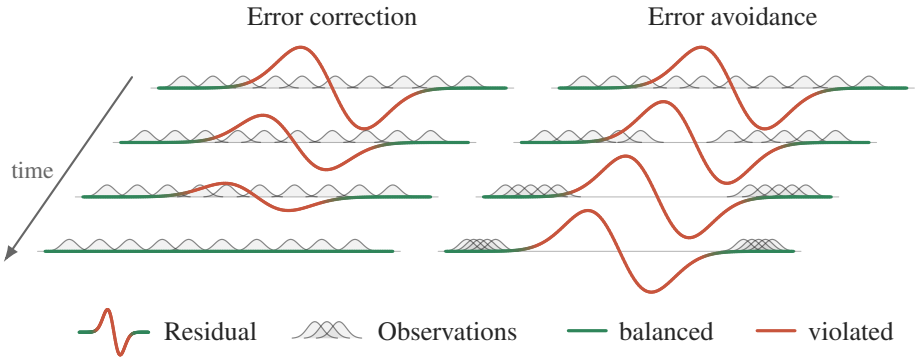

Figure 2: **Two ways to reduce the expected loss.** Error correction (semi-gradient) shrinks the residual under fixed observations until it is balanced. Error avoidance (score term) leaves the residual unchanged and moves the observations onto balanced regions.

3.2 Training dynamics of a GFlowNet and its sampler

Two-player	**Tied single-policy**
Min-min	Full gradient
Min-max	?

To understand how this additional signal can be used constructively, we examine the training dynamics that arise when the GFlowNet and its sampler are treated separately. The semi-gradient and score mechanisms identified above also emerge naturally when the GFlowNet and its sampler are treated as separate components. Let θ parameterize the GFlowNet and ϕ the sampling policy Q_ϕ, and suppose that both minimize the same expected balance loss,

$$\min_{\theta,\phi} \mathbb{E}_{\tau\sim Q_\phi}[L_\theta(\tau)] \implies \dot{\theta} = -\mathbb{E}_{Q_\phi}[\nabla_\theta L_\theta(\tau)], \;\; \dot{\phi} = -\mathbb{E}_{Q_\phi}[L_\theta(\tau)\nabla_\phi \log Q_\phi(\tau)], \tag{8}$$

where the dynamics above correspond to using continuous-time gradient descent to solve the min.

The two dynamics expose separately what is coupled in the full gradient. The GFlowNet corrects balance errors, while the sampler changes which errors are observed. Tying their parameters again, $\phi = \theta$, merges these two effects and recovers the full-gradient update. An adaptive sampler, however, should guide the GFlowNet toward regions where balance violations remain, rather than systematically avoiding them. Rather than jointly minimizing the same expected loss, it is therefore more natural to consider a *min-max* interaction in which the GFlowNet minimizes the balance error while the sampler seeks trajectories where that error remains large (Bengio et al., 2023):

$$\min_{\theta}\max_{\phi} \; \mathbb{E}_{\tau\sim Q_\phi}[L_\theta(\tau)] \implies \dot{\theta} = -\mathbb{E}_{Q_\phi}[\nabla_\theta L_\theta(\tau)], \;\; \dot{\phi} = +\mathbb{E}_{Q_\phi}[L_\theta(\tau)\nabla_\phi \log Q_\phi(\tau)]. \tag{9}$$

Relative to the joint minimization, the GFlowNet update is unchanged, while the sampler direction is reversed and now drives sampling toward unresolved balance violations. Thus, in the decoupled system, turning the score mechanism from error-avoiding to error-seeking requires only reversing the direction of the sampler update. Despite this appealing interpretation, the two-player formulation requires maintaining and training a separately parameterized sampler, introducing additional computational overhead.

3.3 From two players to single-agent dynamics

The joint-minimization dynamics above already suggest a way to collapse the two-player system back into a single set of parameters. After deriving the GFlowNet and sampler updates separately, identifying $\phi = \theta$ makes the two contributions combine exactly into the full gradient of the expected balance loss. This correspondence naturally raises the analogous question for the adversarial dynamics: what single-policy update results from tying the parameters of the min-max game?

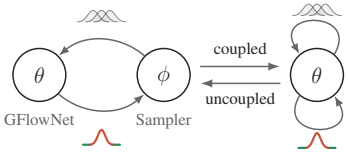

Figure 3: In the uncoupled case, the sampler Q_ϕ samples trajectories and the GFlowNet corrects balance errors, while in the coupled case the GFlowNet assumes both roles.

Applying the identification $\phi = \theta$ to the min-max dynamics completes the correspondence above and yields what we call Naive Negative Reinforce Dynamics (Naive NRD),

$$\dot{\theta}_{\text{NaiveNRD}} = -\mathbb{E}_{Q_\theta}[\nabla_\theta L_\theta(\tau)] + \mathbb{E}_{Q_\theta}[L_\theta(\tau)\nabla_\theta \log Q_\theta(\tau)]. \tag{10}$$

This construction provides the single-policy counterpart of the adversarial GFlowNet–sampler dynamics. However, simply reversing the sampler direction is not sufficient to preserve the desired GFlowNet solution. As we show in Section 5, Naive NRD can admit spurious interior fixed points at which Trajectory Balance remains violated, and therefore need not converge to the reward-proportional distribution. The adversarial construction thus provides the desired error-seeking sampling mechanism, but an additional ingredient is required to make this feedback consistent with the balance equations. This leads to our method, Negative Reinforce Dynamics.

4 Negative Reinforce Dynamics

The adversarial construction from the previous section suggests that an adaptive sampler should direct training toward unresolved balance violations. When the GFlowNet and sampler are separately parameterized, this interpretation is natural. The sampler searches for trajectories with large errors, while the GFlowNet is updated to correct them. However, once their parameters are tied, changing the sampling policy also changes the same forward policy whose balance violations are being measured. To understand why this coupling matters, recall that Trajectory Balance is satisfied on a trajectory $\tau \to x$ when

$$Z_\theta P_F^\theta(\tau) = R(x) P_B^\theta(\tau \mid x). \tag{11}$$

Away from balance, this equality can be violated in either direction. A negative residual corresponds to

$$\delta_\theta(\tau) < 0 \quad \Longleftrightarrow \quad Z_\theta P_F^\theta(\tau) < R(x) P_B^\theta(\tau \mid x), \tag{12}$$

indicating that the forward policy assigns insufficient mass to the trajectory relative to the current balance relation. We refer to this case as under-allocation, and to the opposite inequality as over-allocation. This trajectory-level interpretation also extends to terminal probabilities. If the negative-residual inequality holds for every trajectory terminating at x, then the terminal object itself is under-allocated relative to the current balance relation. When $Z_\theta = Z^\star$, this corresponds directly to assigning less probability to x than its target value $P^\star(x)$.

When the forward policy itself is used for sampling, this distinction becomes consequential. Increasing the probability of a negative-residual trajectory can help expose a region that currently receives insufficient mass. For a positive residual, however, the same update increases the probability of a trajectory that is already over-allocated. This creates a positive feedback loop in which the violation leads to more samples, which in turn reinforce the same violation as illustrated in Figure 4.

Our goal is therefore to construct the GFlowNet–sampler dynamics so that, after their parameters are tied, the useful error-seeking behavior of the adversarial sampler is retained without introducing this feedback loop. The squared Trajectory Balance loss $L_\theta(\tau) = \delta_\theta(\tau)^2$, however, removes precisely the directional information needed to distinguish these two cases. Naive NRD therefore applies the same adversarial feedback to positive and negative residuals. For compactness, we denote the negative-residual contribution by $L_\theta^-(\tau) = L_\theta(\tau)\mathbf{1}\{\delta_\theta(\tau) < 0\}$.

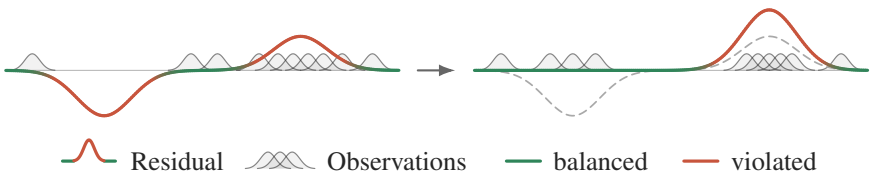

Figure 4: **Naive NRD over-allocation feedback loop.** Reinforcing high-loss trajectories raises their probability regardless of the residual sign. The under-allocated trough ($\delta_\theta < 0$) receives mass until it is balanced, whereas the over-allocated peak ($\delta_\theta > 0$) grows and attracts even more samples. Left: before the update; right: after, with the previous residual dashed.

We incorporate this distinction directly into the two-player construction. The GFlowNet continues to minimize the full Trajectory Balance loss, while the sampler maximizes only the contribution from trajectories with negative residuals. The corresponding simultaneous dynamics are

$$\dot{\theta} = -\mathbb{E}_{Q_\phi}[\nabla_\theta L_\theta(\tau)], \qquad \dot{\phi} = +\mathbb{E}_{Q_\phi}\big[L_\theta^-(\tau)\nabla_\phi \log Q_\phi(\tau)\big]. \tag{13}$$

The resulting sampler dynamics use the magnitude of the balance violation while retaining its direction. Trajectories with large negative residuals receive stronger sampling pressure, whereas positive-residual trajectories do not contribute to the sampler update. As with Naive NRD, we now identify the GFlowNet and sampler parameters. This yields our method, *Negative Reinforce Dynamics* (NRD):

$$\dot{\theta}_{\mathrm{NRD}} = -\mathbb{E}_{Q_\theta}[\nabla_\theta L_\theta(\tau)] + \mathbb{E}_{Q_\theta}\big[L_\theta^-(\tau)\nabla_\theta \log Q_\theta(\tau)\big]. \tag{14}$$

NRD therefore preserves the standard Trajectory Balance correction while using the score mechanism only to reinforce under-allocated trajectories. As we show in the next section, this sign-selective modification is sufficient to recover the balanced solution in the settings we analyze, while the naive sign reversal of Naive NRD can admit spurious interior fixed points. The contrasting behaviors are illustrated in Figure 1.

5 Theoretical Analysis

We analyze the single-policy dynamics obtained after tying the GFlowNet and sampler parameters, focusing on how the change from semi-gradient TB to NRD affects training. Throughout, we consider exact-expectation continuous-time dynamics. Our most general result establishes identical first-order behavior near balance under any differentiable parameterization: the NRD correction is quadratic in the residual and does not change the linearized dynamics. To understand its effect away from balance, we then adopt simpler trajectory-level and one-dimensional models for the sake of analysis, where fixed points, convergence, and hitting times can be characterized more explicitly.

Proposition 1 (Same first-order dynamics near balance)**.** *Let $\theta^\star$ be a balanced solution under a differentiable parameterization, with $\delta_{\theta^\star}(\tau) = 0$ for every trajectory in the sampler's support. Then*

$$\left.\frac{\partial \dot{\theta}_{\mathrm{NRD}}}{\partial \theta}\right|_{\theta^\star} = \left.\frac{\partial \dot{\theta}_{\mathrm{TB}}}{\partial \theta}\right|_{\theta^\star} = -2\,\mathbb{E}_{\tau\sim Q_{\theta^\star}}\left[\nabla_\theta \delta_\theta(\tau)\nabla_\theta \delta_\theta(\tau)^\top\big|_{\theta=\theta^\star}\right]. \tag{15}$$

Any acceleration must therefore arise beyond this shared linearization. To study that regime, we consider a finite set of complete trajectories with positive rewards, a fixed backward policy, free logits for the forward trajectory probabilities p_i, and a jointly learned normalizer Z. The target is $p_i^\star = R(x_i)P_B(\tau_i \mid x_i)/Z^\star$, and trajectories are collected from the trajectory-level mixture $Q_\varepsilon(\tau_i) = (1-\varepsilon)p_i + \varepsilon\mu_i$, where $\varepsilon \in [0,1]$ controls exploration and μ is a fixed distribution with full support. Even under this off-policy mixture, sign-selective feedback preserves the target equilibrium.

Theorem 1 (Fixed-point equivalence)**.** *For every $\varepsilon \in [0,1]$, NRD and semi-gradient Trajectory Balance share a unique interior equilibrium, given by the target distribution $p^\star$ and normalizer $Z^\star$.*

Thus, NRD avoids the additional interior equilibria that can arise with the full expected-loss gradient or Naive NRD. To examine progress toward the target, we measure the spread of the trajectory log-ratios: writing $r_i = \log(p_i/p_i^\star)$, define $D(p,p^\star) = \max_i r_i - \min_i r_i$. The normalizer adds the same offset to all residuals, so the trajectories attaining these extrema also attain the largest and smallest residuals. The NRD correction cannot increase their spread, yielding the following pointwise comparison.

Theorem 2 (Pointwise discrepancy comparison)**.** *For every $\varepsilon \in [0,1]$, the discrepancy obeys $\dot{D}_{\mathrm{NRD}} \le \dot{D}_{\mathrm{TB}}$ when both vector fields are evaluated at the same interior state.*

At a given distribution and normalizer, NRD therefore reduces the discrepancy at least as fast as TB whenever TB is contracting it. This compares the methods at the same state, rather than after they have followed different training paths, and does not by itself guarantee contraction under arbitrary off-policy sampling. On-policy, TB contracts D whenever $D > 0$, so NRD inherits this property. The contraction keeps trajectory probabilities away from the boundary and yields convergence of both the distribution and the normalizer. For the trajectory-level mixture, the comparison is additionally strict when $0 \le \varepsilon < 1$ and the normalizer stays at its conditional stationary value, $\mathbb{E}_{Q_\varepsilon}[\delta] = 0$.

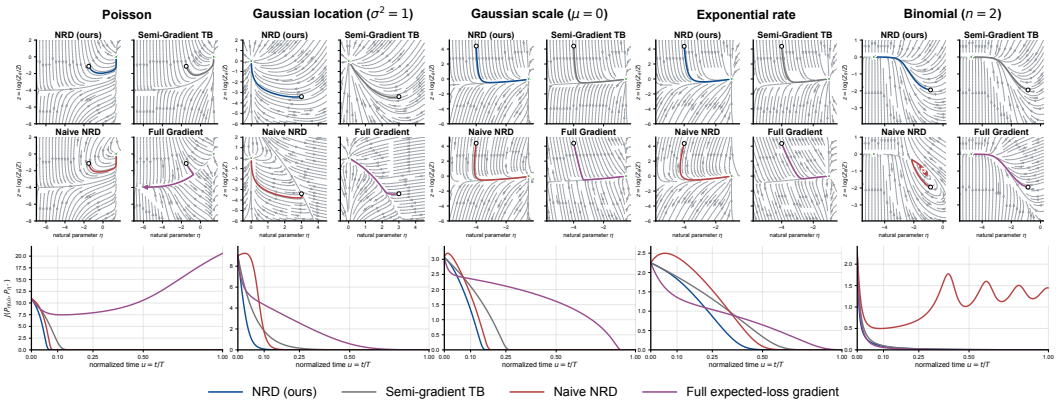

Figure 5: **Population dynamics in exponential families.** Top: phase portraits for Poisson, Gaussian location, Gaussian scale, exponential-rate, and binomial models. Bottom: Jeffreys divergence from the target along the illustrated trajectories. Time is normalized as $u = t/T \in [0, 1]$ and displayed on a $\log(1 + u)$ axis. These examples illustrate faster progress for NRD than semi-gradient TB and possible failure of Naive NRD or the full expected-loss gradient.

Corollary 1 (On-policy convergence)**.** *For $\varepsilon = 0$, NRD converges from every interior initialization to $p(t) \to p^\star$ and $Z(t) \to Z^\star$. At every common interior state away from balance, $\dot{D}_{\mathrm{NRD}} \le \dot{D}_{\mathrm{TB}} < 0$.*

To compare the time needed to approach the target, we next consider a regular one-dimensional exponential family, $p_\eta(\tau) = h(\tau) \exp\{\eta T(\tau) - A(\eta)\}$, with nonconstant sufficient statistic T and target parameter $\eta^\star$. Under on-policy sampling and fast normalization, the NRD correction reinforces the direction of the TB update. Here we measure progress by the Jeffreys divergence $J(p, p^\star) = D_{\mathrm{KL}}(p\|p^\star) + D_{\mathrm{KL}}(p^\star\|p)$, which remains finite between interior parameters even when T is unbounded.

Theorem 3 (Hitting-time advantage)**.** *Start NRD and semi-gradient Trajectory Balance from the same $\eta_0 \neq \eta^\star$, and let $T_{\mathrm{NRD}}(\epsilon)$ and $T_{\mathrm{TB}}(\epsilon)$ be the first times at which $J(p_\eta, p_{\eta^\star}) \le \epsilon$. Then $T_{\mathrm{NRD}}(\epsilon) < T_{\mathrm{TB}}(\epsilon)$ for every $0 < \epsilon < J(p_{\eta_0}, p_{\eta^\star})$.*

The two-trajectory model is the Bernoulli specialization of this result. Its dynamics admit a closed form, and the additive hitting-time advantage can grow without bound as the initial divergence increases. Together, these results distinguish unchanged first-order behavior near balance from stronger progress away from equilibrium, with a strict hitting-time advantage under the one-dimensional assumptions. Figure 5 illustrates population dynamics in representative exponential families; proofs and the two-trajectory specialization appear in Appendix D, with the latter in Appendix D.10.

6 Experiments

Our experiments test whether the behavior predicted by the theory holds in practice. We ask whether NRD improves distribution learning and high-reward discovery relative to TB, and whether these gains require a separately trained exploratory policy. Comparing with alternative training losses isolates the effect of the update rule; comparing with auxiliary exploration methods tests how much exploration benefit survives when sampler and GFlowNet are tied. We evaluate both target-distribution fit and the discovery of distinct high-reward modes.

6.1 Experimental setup

For distribution learning, we use Rings, Moons, and 8 Gaussians in DiffusionGrid, adapted from the finite-horizon random-walk setting of Dall'Antonia et al. (2026b), and the HyperGrid construction benchmark (Bengio et al., 2021; Whitammer et al., 2022). Many paths lead to each terminal state, but finite-budget exploration can still concentrate on only part of the multimodal target. We measure estimated terminal total variation (TV) from normalized reward. For high-reward discovery, we use

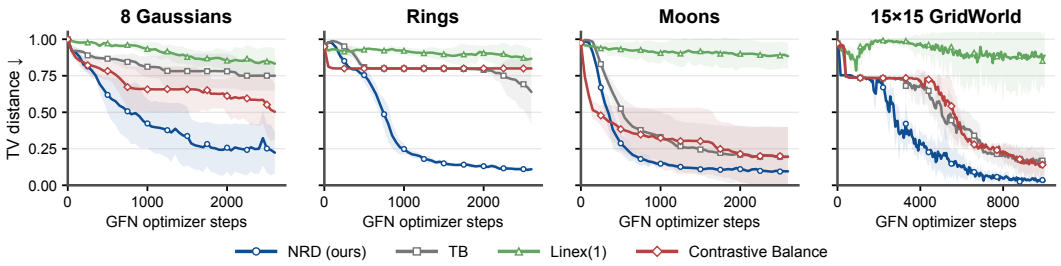

Figure 6: **Distribution learning with alternative training objectives.** Estimated terminal TV versus GFlowNet optimizer steps for NRD, TB, Linex(1), and Contrastive Balance. Lines and shaded bands show the mean and one standard deviation over four seeds. The GridWorld panel corresponds to the 15×15 HyperGrid. Lower values are better.

Bags, Knapsack128, and autoregressive sequence tasks adapted from prior structured benchmarks (da Silva et al., 2025), tracking the mean log-reward of the top 200 unique objects encountered. We also study peptide design (Jain et al., 2022; Dall'Antonia et al., 2026a), counting distinct sequences that satisfy a screening criterion from an oracle trained on GRAMPA (Witten & Witten, 2019).

The first baseline group changes the training objective without adding an exploration policy. It comprises TB, the asymmetric Linex(1) loss (Hu et al., 2025), and the pairwise Contrastive Balance (CB) objective (da Silva et al., 2024). The second group explicitly trains exploratory models through ACE's adversarial sampler (Dall'Antonia et al., 2026a), SA-GFN's novelty-seeking sibling policy (Madan et al., 2025), or Adaptive Teacher's discrepancy-driven teacher (Kim et al., 2025).

We use four seeds and the prescribed budgets. Each such step counts a policy update, including auxiliary-policy updates. Appendix A gives environment definitions, architectures, optimization settings, estimators, and the complete structured results. Appendix B provides baseline context and wall-time curves; Appendix C contains the score-distribution ablation.

6.2 Distribution Learning and Exploration

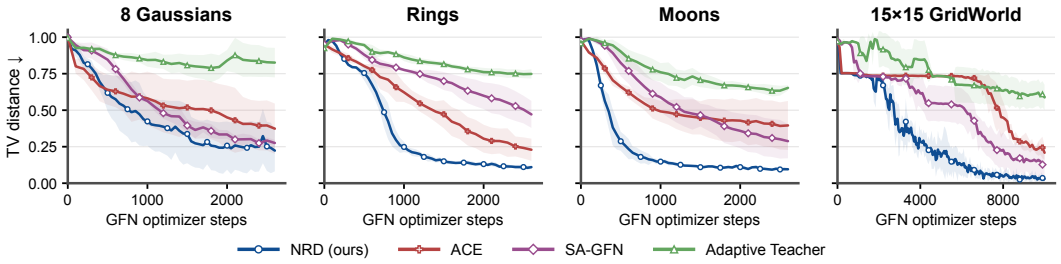

Figure 7: **Comparison with explicit exploration mechanisms.** NRD versus ACE, SA-GFN, and Adaptive Teacher on the same four distribution-learning tasks. Curves use the full optimizer-step budgets and report means with one-standard-deviation bands across four seeds. NRD has the lowest mean final estimated TV in each panel.

Figures 6 and 7 address the same exploration problem from two directions. First, can a single-policy update escape the poor distribution fits observed under other GFlowNet losses? Rings provides the clearest example. TB and CB remain at high error for substantial portions of training, whereas NRD keeps improving, ending at estimated TV 0.110 against TB's 0.639. This illustrates the gap between a loss admitting the correct distribution and its optimization reaching that distribution within a finite budget. Across the four tasks, NRD has the lowest mean estimated TV.

Second, is an independently trained exploration policy necessary for this improvement? Figure 7 shows that auxiliary methods can substantially reduce the distributional error seen in the first comparison, but NRD remains competitive without a second sampler and has the lowest estimated TV in each panel. SA-GFN is particularly competitive on 8 Gaussians, with overlapping seed

variability. Together, these comparisons support the proposed connection to adaptive exploration, with useful exploration pressure surviving when the GFlowNet is responsible for correction balance and sampling trajectories rather than requiring an additional policy.

The optimizer-step comparison includes auxiliary-policy updates, while the wall-time views in Appendix B expose the recorded cost of training and online evaluation. The advantage on Rings and Moons also appears over substantial shared portions of those time axes. The structured results below further qualify the conclusion by showing cases where auxiliary exploration finds higher-reward objects despite NRD's stronger distribution-learning results.

6.3 Exploration with sparse rewards

We next ask whether the exploration benefit translates into discovering distinct peptide candidates among sequences of up to ten amino acids. The 19-amino-acid alphabet already allows $19^{10} \approx 6.1 \times 10^{12}$ full-length candidates. Figure 8 accumulates oracle-qualified sequences sampled from the principal policy without exploration noise; repeated discoveries do not increase the count. A sequence qualifies when the maximum predicted activity probability across five random forests is at least 0.95. The evaluation schedule, oracle, and aggregation protocol are specified in Appendix A.4. This measures the breadth of computationally screened discoveries, not global distribution matching or experimentally validated biological activity.

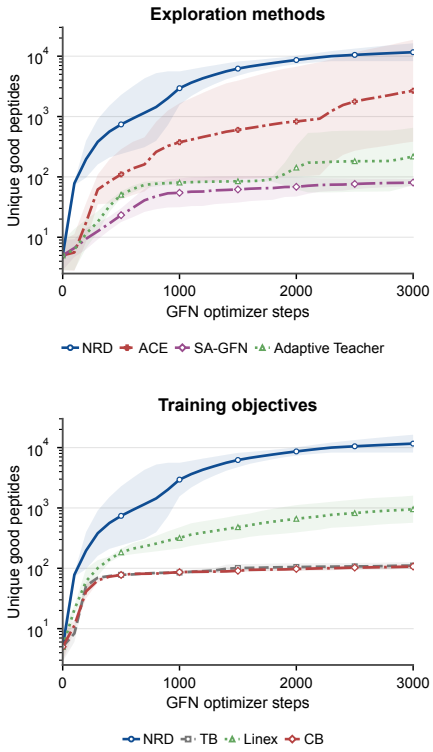

Figure 8: **Cumulative peptide discovery.** Auxiliary exploration methods above; alternative objectives below. Geometric means over four seeds.

At the end of the budget, NRD discovers a geometric mean of $11{,}637$ distinct qualifying peptides, compared with $2{,}674$ for ACE, 953 for Linex, 218 for Adaptive Teacher, 111 for TB, 107 for CB, and 81 for SA-GFN. The aggregate advantage is approximately $4.35\times$ over ACE and $105\times$ over TB. The trajectory of the count indicates that this advantage develops during training, while the per-seed results show a narrower multiplicative spread than ACE. The advantage is not seed-wise dominance, since ACE exceeds NRD in one of the four seeds.

For the remaining structured tasks, we assess the quality of the discoveries accumulated during training through the mean log-reward of the top 200 unique objects. The complete results in Table 2 of Appendix A show that NRD attains higher final means than TB on Knapsack and all sequence tasks, while the Bags results are comparable within the observed seed variability. NRD also remains competitive with auxiliary exploration methods, including those driven by auxiliary rewards, although some sequence tasks favor ACE or Adaptive Teacher. These results extend the peptide findings to other combinatorial spaces, supporting effective high-reward discovery without a separately trained exploration policy.

7 Conclusion

We introduced Negative Reinforce Dynamics, a novel GFlowNet training scheme that arises as a 'weight-tied' version of an adversarial game between the GFlowNet and a trajectory sampler. By selectively reinforcing under-allocated trajectories, NRD incorporates the adversarial sampler's exploratory feedback into the model's own update, greatly enhancing discovery with little additional cost. Our experiments support the view that these exploratory benefits can survive parameter tying, showing improved distribution learning and high-reward discovery relative to Trajectory Balance, alongside competitive performance against auxiliary exploration methods. NRD thus provides a way to exploit the adversarial formulation without maintaining a separately trained sampler.

AI USE STATEMENT

Generative AI tools were used to provide feedback on the research methodology and experimental evaluation, and to assist with the interpretation and presentation of experimental results. They were also used for literature search and summarization, as well as for drafting, editing, and improving the clarity and organization of the manuscript. The core research ideas, mathematical formulation, theoretical claims, and proofs were developed and verified by the authors. All AI-assisted suggestions and generated text were reviewed and revised by the authors, who take responsibility for the final content of the paper.

REFERENCES

Brian Bartoldson, Siddarth Venkatraman, James Diffenderfer, Moksh Jain, Tal Ben-Nun, Seanie Lee, Minsu Kim, Johan Obando-Ceron, Yoshua Bengio, and Bhavya Kailkhura. Trajectory balance with asynchrony: Decoupling exploration and learning for fast, scalable llm post-training, 2025. URL https://arxiv.org/abs/2503.18929.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27381–27394, 2021.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. GFlowNet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023.

Tiago da Silva, Luiz Max Carvalho, Amauri Souza, Samuel Kaski, and Diego Mesquita. Embarrassingly parallel gflownets. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 45406–45431. PMLR, 2024. URL https://proceedings.mlr.press/v235/silva24a.html.

Tiago da Silva, Rodrigo Barreto Alves, Eliezer de Souza da Silva, Amauri Souza, Vikas Garg, Samuel Kaski, and Diego Mesquita. When do GFlowNets learn the right distribution? In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9GsgCUJtic.

Pedro Dall'Antonia, Tiago da Silva, Daniel Csillag, Salem Lahlou, and Diego Mesquita. Avoid what you know: Divergent trajectory balance for GFlowNets. *arXiv preprint arXiv:2602.17827*, 2026a. URL https://arxiv.org/abs/2602.17827.

Pedro Dall'Antonia, Tiago da Silva, Daniel Augusto de Souza, César Lincoln C. Mattos, and Diego Mesquita. Boosted gflownets: Improving exploration via sequential learning, 2026b. URL https://arxiv.org/abs/2511.09677.

Tristan Deleu, Mizu Nishikawa-Toomey, Jithendaraa Subramanian, Esmeralda S. Whitammer, Laurent Charlin, and Yoshua Bengio. Joint bayesian inference of graphical structure and parameters with a single generative flow network. In *Advances in Neural Information Processing Systems 36*, NeurIPS 2023, pp. 31204–31231. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-1360. URL http://dx.doi.org/10.52202/075280-1360.

Edward J. Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Esmeralda S. Whitammer. Amortizing intractable inference in large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.04363.

Rui Hu, Yifan Zhang, Zhuoran Li, and Longbo Huang. Beyond squared error: Exploring loss design for enhanced training of generative flow networks. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.02596.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with GFlowNets. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9786–9801. PMLR, 2022.

Minsu Kim, Sanghyeok Choi, Taeyoung Yun, Emmanuel Bengio, Leo Feng, Jarrid Rector-Brooks, Sungsoo Ahn, Jinkyoo Park, Esmeralda S. Whitammer, and Yoshua Bengio. Adaptive teachers for amortized samplers. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=BdmVgLMvaf.

Kanika Madan, Jarrid Rector-Brooks, Maksym Korablyov, Emmanuel Bengio, Moksh Jain, Andrei Cristian Nica, Tom Bosc, Yoshua Bengio, and Esmeralda S. Whitammer. Learning gflownets from partial episodes for improved convergence and stability. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23467–23483. PMLR, 2023. URL https://proceedings.mlr.press/v202/madan23a.html.

Kanika Madan, Alex Lamb, Emmanuel Bengio, Glen Berseth, and Yoshua Bengio. Towards improving exploration through sibling augmented GFlowNets. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HH4KWP8RP5.

Esmeralda S. Whitammer, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in GFlowNets. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-0431.

Esmeralda S. Whitammer, Salem Lahlou, Tristan Deleu, Xu Ji, Edward Hu, Katie Everett, Dinghuai Zhang, and Yoshua Bengio. Gflownets and variational inference. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=uKiE0VIluA-.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992. doi: 10.1007/BF00992696.

Jacob Witten and Zack Witten. Deep learning regression model for antimicrobial peptide design. *bioRxiv*, 2019. doi: 10.1101/692681. URL https://doi.org/10.1101/692681.

A Reproducibility and complete structured-discovery results

A.1 Evaluation and optimization budgets

We use seeds 39–42 and retain the prescribed final budget for each experiment, without selecting a best checkpoint. Distribution-learning curves and the structured-discovery tables report arithmetic means and sample standard deviations across these four seeds. Peptide counts instead use geometric means and one population standard deviation in log-count space. These bands describe seed variability, not confidence intervals. An optimizer step in the grid and peptide comparisons counts a GFlowNet-policy update: TB, NRD, Linex, and CB contribute one per training iteration, whereas ACE, SA-GFN, and Adaptive Teacher contribute two, including their auxiliary policy. The separate predictor update used for SA-GFN's intrinsic reward is not a GFlowNet-policy update. Accordingly, the DiffusionGrid budget is 2,600 iterations for the single-policy methods and 1,300 for auxiliary-policy methods; HyperGrid uses 10,000 and 5,000, respectively, and peptides use 3,000 and 1,500. The structured tasks retain their original iteration budgets, listed in Table 1, for every method.

The grid metric is estimated terminal total variation, $\widehat{\mathrm{TV}} = \frac{1}{2}\sum_x |\hat{p}_\theta(x) - p^\star(x)|$, where $p^\star(x) = R(x)/\sum_y R(y)$. Both terminal grids are enumerated, but the forward marginal is reconstructed from ten backward trajectories per terminal state. Specifically, samples from $P_B(\cdot \mid x)$ estimate $P_T(x) = \mathbb{E}_{P_B(\cdot|x)}[P_F(\tau)/P_B(\tau \mid x)]$; the resulting nonnegative weights are normalized over the terminal grid. Thus these curves are Monte Carlo estimates, not exact forward-policy marginals. Evaluations are recorded every 50 training iterations; the axes convert iterations to the policy-update count above. For structured discovery, a history queue retains the 200 highest-reward unique terminal objects seen during training. The reported statistic is the arithmetic mean of their log rewards, not the reward of a fresh evaluation batch.

A.2 Distribution-learning environments and models

DiffusionGrid adapts the finite-horizon lazy random walk of Dall'Antonia et al. (2026a), with Rings and 8 Gaussians targets and an additional Moons target. It starts at the origin of $\{-15, \dots, 15\}^2$ and takes 30 actions. Each action moves one unit in either coordinate direction or stays in place; boundary-crossing actions are masked. Time is part of the state, so spatial revisits still define an acyclic construction process. Before boundary masking, the horizon admits 5^{30} action sequences for only 31^2 terminal locations, making path multiplicity substantial. The Rings density is the sum of two radial Gaussian kernels with radii 3 and 12, radial standard deviation 1, and a common multiplier of 10. The 8 Gaussians density sums eight isotropic Gaussian kernels of standard deviation 1, centered at equally spaced points on a circle of radius 12. Moons sums kernels of standard deviation 1 over 256 equally spaced anchors on each of two semicircles: their radius is 9, horizontal shifts are -4.5 and 4.5, and the lower semicircle is shifted down by 2.7. For each density $d(x)$, the reward is $R(x) = e[(1 - 10^{-6})d(x) + 10^{-6}]$, up to a negligible numerical floor inside the kernel sum. The common reward exponent is one.

The principal DiffusionGrid forward policy has two hidden layers of width 128 with ReLU activations and five output logits. The separately learned backward policy is a linear map to five logits. Both receive spatial coordinates, normalized time, and sine/cosine time features at eight frequencies 2^k, $k = 0, \dots, 7$. There is no shared forward–backward backbone. The ACE auxiliary networks use one hidden layer of width 64, with 16 forward and eight backward time frequencies. Other auxiliary policies follow the principal-policy dimensions. Training uses a constant action-wise exploration probability $\varepsilon = 0.1$ and base batch size 1,024. TB, NRD, CB, and the auxiliary-policy baselines use AdamW with policy learning rates 0.005, normalizer learning rate 0.05 where applicable, and weight decay 0.01. The plotted Linex configuration uses AdamW with forward/backward learning rates 0.001 and normalizer learning rate 0.1. These rates are constant, without a learning-rate schedule or gradient clipping.

HyperGrid follows the coordinate-increment construction of Bengio et al. (2021); Whitammer et al. (2022), with states $x \in \{0, \dots, 14\}^2$. Starting at $(0, 0)$, the policy increments one coordinate or terminates, with invalid increments masked. Let $a_i = |2x_i/14 - 1|$. Its reward is $R(x) = 10^{-3} + 3 \prod_{i=1}^{2} \mathbf{1}\{0.6 < a_i < 0.8\}$. The policy receives the coordinate vector and uses three hidden layers of width 128 with leaky-ReLU activations, followed by three action logits. Its backward policy is fixed and uniform over valid parents. Batch size is 512 and $\varepsilon = 0.1$. Matrix-valued policy parameters use Muon and non-matrix parameters use its Adam-based branch, without weight decay or gradient clipping. The network learning rate increases linearly from zero to 0.01 over the first 10% of iterations, then follows a cosine decay to 0.0001 at the budget endpoint. The Linex curve uses the same optimizer and schedule with peak rate 0.001 and final rate 0.00001. The learned scalar $\log Z$ is initialized at zero and updated with Adam at constant rate 0.1. DiffusionGrid also initializes $\log Z$ at zero; CB eliminates the normalizer from its paired-residual objective.

A.3 Structured environments, optimization, and full results

The structured tasks adapt the bag, quadratic-knapsack, and autoregressive sequence constructions used by Dall'Antonia et al. (2026a); the sequence family also follows da Silva et al. (2025). The definitions below specify our instances and rewards rather than assuming identical published configurations. Bags18 and Bags24 construct multisets of exactly 12 items from 18 and 24 item types, respectively; repeated items are permitted. A state is its count vector, and each action adds one item. With positive lognormal utilities u_i generated using reward seed 43, the log reward is $\log R(x) = 10[\log(u^\top x) - c]$, where $c = \log\lfloor 12 \max_i u_i/2 \rfloor$ is the common normalizing offset. The backward policy removes one item type uniformly among types with positive count, rather than sampling individual copies proportionally to their counts.

Knapsack128 constructs a subset of 128 items, with each item selectable at most once and total weight at most 60. Integer weights are sampled uniformly from $\{2, \dots, 58\}$ for each run seed and shared by methods at that seed. Construction stops when no remaining item fits. The reward is $R(x) = u^\top x + \frac{1}{2} x^\top A x$. Using reward seed 43, the utilities are uniform on $[0.1, 1)$, and the symmetric interaction matrix is formed from standard-normal off-diagonal entries with sparsity 0.15. Each diagonal entry is the absolute off-diagonal row sum plus 0.1; entries are then divided by

$\sqrt{w_i w_j} + 10^{-6}$. The backward policy removes a selected item uniformly. The weight realization varies across seeds, while methods within a seed face the same instance.

The sequence environments append tokens autoregressively to a fixed length L, so the backward transition is deterministic. Binary32 uses $L = 32$ and vocabulary size $V = 2$; the remaining tasks use each combination of $L \in \{24, 32\}$ and $V \in \{4, 6\}$. All five tasks use the additive log reward $\log R(x) = \sum_{i=1}^{L}(u_{x_i} + v_i)$, with token utilities u and position offsets v sampled uniformly from $[0, 1)$ using reward seed 43. In particular, Binary32 here uses this reward, not a separate Hamming-distance-to-modes objective. Network inputs are zero-padded prefix vectors for sequences and item-count vectors for Bags and Knapsack.

All structured-task forward policies use three hidden layers of width 128 with leaky-ReLU activations (negative slope 0.01), followed by action logits. Backward policies are fixed as described above. The policy optimizer, 10%-warmup cosine schedule, scalar normalizer initialization, and separate normalizer learning rate are those of the standard HyperGrid configuration. Thus the policy peak learning rate is 0.01 and the constant Adam rate for $\log Z$ is 0.1; no weight decay or gradient clipping is applied. Auxiliary policies use the same architecture and base learning rates. The exploration probability is constant at 0.1, mixing with a uniform distribution over valid actions at each step. ACE and SA-GFN split the configured base batch equally between their two rollout branches, while TB, NRD, and Adaptive Teacher use the full base batch per rollout.

Table 1: Structured-task configurations. Each budget is a number of training iterations for every method; checkpoint intervals use the same unit.

Environment	Size / vocabulary	Batch	Iterations	Checkpoint interval
Bags18	$18/12$	32	1,500	50
Bags24	$24/12$	32	1,500	50
Knapsack128	$128/60$	32	256	10
Binary32	$32/2$	64	2,000	50
Seq24/V4	$24/4$	64	2,000	50
Seq24/V6	$24/6$	64	2,000	50
Seq32/V4	$32/4$	64	2,000	50
Seq32/V6	$32/6$	64	2,000	50

Table 2: Complete structured-discovery results at the budgets in Table 1. Entries are the mean log reward of the top 200 unique objects, averaged over four seeds, $\pm$ one sample standard deviation. Higher is better; bold marks means whose ± 1 SD intervals overlap that of the highest mean, using unrounded values. This describes seed variability, not statistical equivalence.

Environment	NRD	TB	ACE	SA-GFN	Adaptive Teacher
Bags18	$\mathbf{5.734} \pm 0.008$	$\mathbf{5.748} \pm 0.038$	5.641 ± 0.049	$\mathbf{5.726} \pm 0.023$	$\mathbf{5.719} \pm 0.070$
Bags24	$\mathbf{5.459} \pm 0.020$	$\mathbf{5.468} \pm 0.044$	5.260 ± 0.075	$\mathbf{5.395} \pm 0.073$	$\mathbf{5.455} \pm 0.077$
Knapsack128	$\mathbf{3.375} \pm 0.093$	$\mathbf{3.219} \pm 0.098$	$\mathbf{3.235} \pm 0.070$	3.137 ± 0.064	3.191 ± 0.081
Binary32	38.897 ± 0.073	38.483 ± 0.028	$\mathbf{39.010} \pm 0.497$	38.488 ± 0.016	$\mathbf{39.306} \pm 0.292$
Seq24/V4	$\mathbf{31.667} \pm 0.107$	31.248 ± 0.040	$\mathbf{32.105} \pm 0.580$	31.221 ± 0.037	$\mathbf{31.582} \pm 0.045$
Seq24/V6	$\mathbf{30.244} \pm 0.199$	29.781 ± 0.073	$\mathbf{30.252} \pm 0.265$	29.781 ± 0.049	$\mathbf{29.968} \pm 0.020$
Seq32/V4	$\mathbf{40.751} \pm 0.286$	39.640 ± 0.048	$\mathbf{42.011} \pm 1.117$	39.626 ± 0.083	39.865 ± 0.086
Seq32/V6	$\mathbf{38.923} \pm 0.577$	37.504 ± 0.139	$\mathbf{39.997} \pm 0.913$	37.537 ± 0.104	37.539 ± 0.130

Table 2 reports all structured tasks discussed in Section 6.3. TB has the highest mean on both Bags tasks, NRD on Knapsack, Adaptive Teacher on Binary32, and ACE on all four nonbinary sequence tasks. The small mean differences in Bags and Seq24/V6 should be read together with their seed variability; these endpoint rankings are descriptive and are not significance tests.

A.4 Peptide oracle, model, and discovery protocol

Our peptide task follows the oracle-guided sequence-design setting (Jain et al., 2022; Dall'Antonia et al., 2026a), with the alphabet and screening protocol specified here. Peptides are generated

by appending amino acids or an end-of-sequence token, with maximum length ten. The alphabet contains the 19 standard amino acids other than cysteine; the end-of-sequence token also provides padding. The policy embeds each token in 32 dimensions, concatenates an eight-token context window with four sinusoidal position features, and applies one 64-unit ReLU hidden layer before the output logits. The backward transition is deterministic. The base batch is 4,096 and the constant exploration probability is 0.3. NRD, TB, CB, ACE, SA-GFN, and Adaptive Teacher use AdamW with network rate 0.05 and weight decay 0.01; learned log normalizers use rate 0.1. Linex uses Adam, no weight decay, network rate 0.005, and normalizer rate 0.02. There is no learning-rate schedule or gradient clipping. ACE and SA-GFN collect 2,048 trajectories per branch. All methods use the same seed-specific initial principal-policy checkpoint.

The oracle consists of five random-forest classifiers trained on filtered GRAMPA records (Witten & Witten, 2019), one for each of *E. coli*, *S. aureus*, *P. aeruginosa*, *B. subtilis*, and *C. albicans*. The preprocessing retains sequences of length one to ten, excludes cysteine and specified chemical modifications, and removes records from YADAMP. Listed peptides associated with each organism provide positive examples; an equal number of randomly generated sequences, using the empirical positive-length distribution, provide negative examples. The respective positive/negative counts per classifier are 664/664, 653/653, 396/396, 170/170, and 227/227. Classifiers use flattened one-hot sequence features, 500 trees, square-root feature subsampling, balanced class weights, and random seed 42. These models provide computational screening scores, not experimental validation of generated candidates.

Let $p(x)$ be the maximum positive-class probability across the five classifiers, and let $\ell(x)$ be the sequence length. The training log reward is

$$\log R(x) = \max\left\{-30, (\ell(x)+1)\min\left[0, \frac{\operatorname{logit} p(x) - \operatorname{logit}(0.95)}{0.3}\right]\right\}. \tag{16}$$

This transformation is bounded above by zero, with reward one for qualifying sequences; its clipping is part of the reward definition. A discovery requires $p(x) \geq 0.95$. Evaluation samples 1,000 sequences with zero exploration noise at initialization and every 100 GFlowNet-policy updates through 3,000. Sequences are deduplicated cumulatively within each run, including those from the initial evaluation. The reported $11{,}637$ is the geometric mean of these cumulative counts at step 3,000, not a count from a single batch or a sum over seeds. Generation enforces the vocabulary and length bound; no additional biological-validity filter is applied to the discovery count.

B Connections to exploration baselines and wall-time comparisons

B.1 GFlowNet losses without an auxiliary exploration policy

Trajectory Balance (TB) (Whitammer et al., 2022) trains a generative policy by matching the forward trajectory flow to the terminal reward times the backward trajectory probability. Its squared log-residual penalty provides the reference update in our comparisons. Changing this penalty can change the training dynamics without changing the desired balance relation. In particular, Linex(1) (Hu et al., 2025) is asymmetric: with our residual convention, its implemented penalty is $\exp(-\delta) - 1 + \delta$. Negative residuals therefore receive different weights from equally large positive ones. This is distinct from NRD, which retains the TB semi-gradient and adds a detached, negative-residual score correction to the policy update.

Contrastive Balance (da Silva et al., 2024) compares trajectory log-ratios pairwise. The evaluated implementation minimizes squared differences between paired TB residuals, so the shared log-normalizer cancels. Like TB and Linex, it does not train a separate exploration policy. These comparisons distinguish a change in the balance loss from the additional sampling-distribution pressure in NRD; they do not exhaust all possible losses or their hyperparameter settings. Correctness of a balance relation at its solution is also separate from finite-budget coverage: a full-support policy can assign practically negligible probability to important regions long before it reaches that solution.

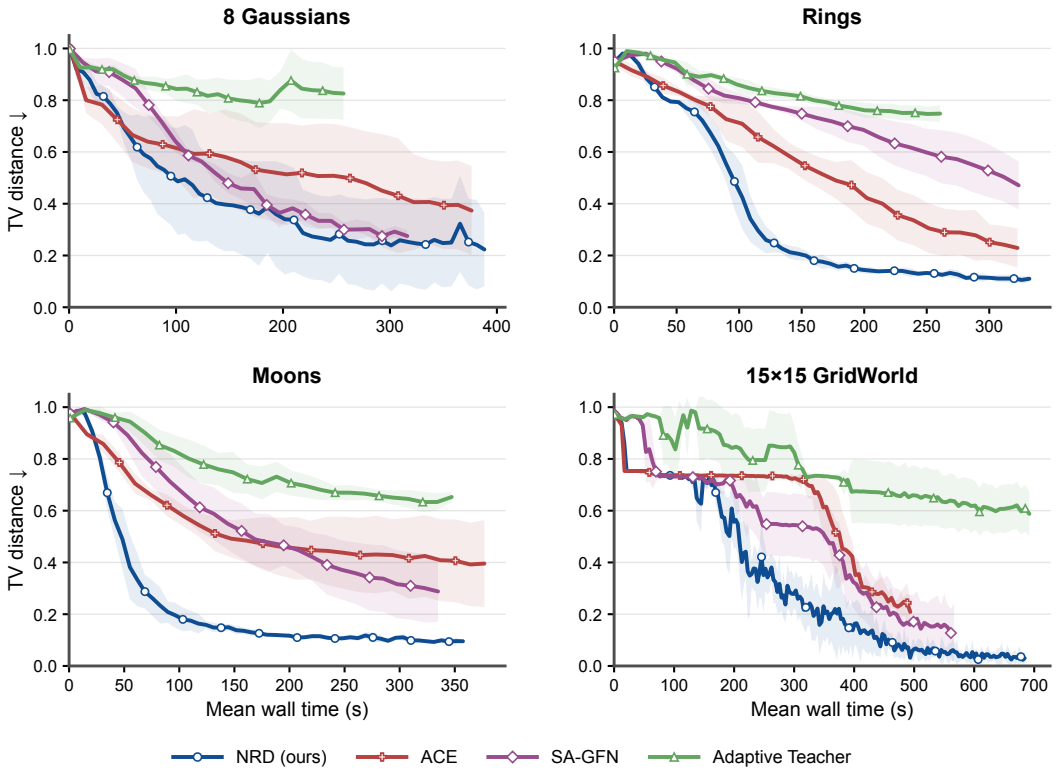

Figure 9: **Distribution learning versus recorded wall time.** Estimated TV for the auxiliary-policy comparison in Figure 7, using the same source runs and seeds 39–42. Each point pairs mean elapsed time with mean TV at the matching checkpoint; shading is one sample standard deviation in TV. Elapsed time includes training and online evaluation. Curves retain their recorded endpoints, which differ across methods. Lower is better.

B.2 Methods with an auxiliary exploration policy

ACE (Dall'Antonia et al., 2026a) couples a learner to an adversarial sampler that seeks trajectories with large learner discrepancies. Its two-player structure is especially relevant here: NRD investigates whether an exploration-inducing update can be retained after identifying sampler and learner parameters. ACE instead maintains the additional sampler throughout training. The comparison tests the practical effect of this distinction, not an equivalence between their finite-step implementations.

SA-GFN (Madan et al., 2025) augments the principal GFlowNet with a sibling policy that uses an intrinsic novelty signal. In the evaluated implementation, random-network-distillation prediction error supplies this signal, and the principal policy learns from sibling-generated trajectories as well as its own. This adds both a policy and a learned novelty predictor. Adaptive Teacher (Kim et al., 2025) instead constructs a teacher reward from the student's trajectory-balance discrepancy and the task reward, emphasizing trajectories on which the student needs improvement. The teacher and student are optimized separately. Both approaches make exploration adaptive, but they require learned components beyond the principal policy, unlike NRD.

B.3 Recorded wall-time comparisons

Figure 9 re-expresses the distribution-learning comparison against elapsed time rather than GFlowNet-policy updates. On Rings and Moons, the lower estimated TV of NRD remains visible over substantial shared portions of the recorded time range. The 8 Gaussians comparison is less separated from SA-GFN and has appreciable seed variability. HyperGrid also shows earlier progress by NRD after the initial plateau, but methods finish at different times. Thus the benefit is not visible only under

the policy-step accounting, while neither the plots nor their unequal endpoints establish a universal wall-time advantage.

Elapsed time is the recorded cumulative wall time, or the sum of cumulative training and online-evaluation time where a wall-time field is unavailable. Aggregation is by checkpoint, not by interpolation onto a common time grid: horizontal positions are the means of the four seed times. The figure therefore describes the recorded error–time trade-off rather than an equal-time stopping rule.

C SENSITIVITY TO THE EXPLORATION DISTRIBUTION

We study how the score distribution affects training when trajectories are collected with an action-wise ε-greedy mixture $Q_\varepsilon(a \mid s) = (1-\varepsilon)P_F(a \mid s) + \varepsilon U(a \mid s)$, where U is uniform over valid actions. The ablation compares the corrections based on $\nabla \log P_F$ and $\nabla \log Q_\varepsilon$, keeping the remaining settings fixed within each environment and noise level. Four seeds are used for each pair, with $\varepsilon \in \{0.1, 0.3, 0.5, 0.8\}$. We summarize entire learning curves rather than only the final checkpoint, since two variants may reach similar final errors while differing substantially in their transients. This ablation is separate from the main baseline comparisons.

For an error curve $m(t)$, we compute the normalized area over its prescribed budget T and report the relative reduction obtained with Q_ε:

$$A_s = \frac{1}{T}\int_0^T \overline{m_s(t)}\, dt, \qquad \Delta_{\mathrm{AUC}} = 100\left(1 - \frac{A_Q}{A_{P_F}}\right). \tag{17}$$

The integral is evaluated by the trapezoidal rule at recorded checkpoints, and the bar denotes the arithmetic mean across seeds. Positive values indicate lower cumulative error with Q_ε; negative values indicate the opposite. The grids use estimated TV. For the structured tasks, we use a batch-normalized implementation of Flow Consistency in Subgraphs (FCS) (da Silva et al., 2025), averaged over evaluation batches: terminal states are sampled by a uniform-action rollout, model and reward weights are normalized within each batch, and their TV is computed. FCS is therefore a consistency proxy, not the exact TV over the entire terminal space and not the historical top-200 reward metric used in Section 6.3.

Table 3: **Percentage reduction in error-curve area from using Q_ε rather than P_F in the score correction.** Positive values favor Q_ε. All entries aggregate four seeds over the full prescribed budget; they are effect sizes, not significance tests. The two blocks use different error metrics.

Environment	$\varepsilon = 0.1$	0.3	0.5	0.8
Estimated terminal TV				
Rings	−11.14	2.50	3.88	−22.96
Moons	−34.92	−2.14	−6.13	−10.25
8 Gaussians	−13.93	14.03	31.43	3.71
HyperGrid	6.30	11.40	−1.86	−0.20
FCS				
Bags18	1.44	5.89	14.12	20.75
Bags24	0.87	8.67	10.89	15.24
Knapsack128	−1.06	−1.88	2.85	8.00
Binary32	42.26	45.50	32.66	7.60
Seq24/V4	17.13	25.96	18.64	6.91
Seq24/V6	6.32	18.87	16.82	5.52
Seq32/V4	23.08	33.09	25.95	8.60
Seq32/V6	16.71	25.06	21.83	6.41

Table 3 shows that the effect depends on both the task and the exploration level. In the structured tasks, the Q_ε correction reduces area in 30 of the 32 environment–noise combinations, with the largest reduction reaching 45.50% on Binary32 at $\varepsilon = 0.3$. Knapsack is the exception at the two lowest nois e levels, where the differences favor P_F by less than two percent. The grid results are more mixed: only seven of the sixteen combinations favor Q_ε, and Moons favors P_F at every evaluated noise level. These observations establish sensitivity to the score distribution and substantial transient benefits in many structured tasks, but do not support a universal ordering of the two variants.

D PROOFS OF THE THEORETICAL RESULTS

This section first establishes the marginalization and gradient identities used in the main text, and then proves the results in Section 5. We use the same notation throughout and introduce additional notation only when needed.

D.1 TRAJECTORY BALANCE IMPLIES THE TARGET TERMINAL MARGINAL

Assume complete forward trajectories have total probability one, and the backward policy defines a normalized distribution over trajectories ending at each terminal object. If Equation 2 holds for every complete trajectory, summing over all trajectories ending at a fixed x gives

$$ZP_T(x) = Z \sum_{\tau \to x} P_F(\tau) = R(x) \sum_{\tau \to x} P_B(\tau \mid x) = R(x). \tag{18}$$

Summing this identity over x yields $Z = \sum_x R(x) = Z^\star$, because $\sum_x P_T(x) = 1$. Substitution gives $P_T(x) = R(x)/Z^\star$ for every terminal object. This is the usual marginalization consequence of trajectory balance (Whitammer et al., 2022; Bengio et al., 2023).

D.2 SCORE-FUNCTION DECOMPOSITION OF THE EXPECTED-LOSS GRADIENT

For Equations 1 and 6, assume a parameter-independent trajectory support on which Q_θ is positive and differentiable, and that derivatives can be interchanged with the expectation. On a finite support, the product rule directly gives

$$\begin{aligned} \nabla_\theta \sum_\tau Q_\theta(\tau) L_\theta(\tau) &= \sum_\tau Q_\theta(\tau) \nabla_\theta L_\theta(\tau) + \sum_\tau L_\theta(\tau) \nabla_\theta Q_\theta(\tau) \\ &= \mathbb{E}_{Q_\theta}[\nabla_\theta L_\theta] + \mathbb{E}_{Q_\theta}[L_\theta \nabla_\theta \log Q_\theta], \end{aligned} \tag{19}$$

where the second line uses $\nabla_\theta Q_\theta = Q_\theta \nabla_\theta \log Q_\theta$. The same argument applies to densities when differentiation under the integral is justified. This is the classical score-function identity (Williams, 1992); the semi-gradient retains only its first term.

D.3 THE TB SEMI-GRADIENT AND REVERSE KL

We derive Equation 7, following the identity established by Whitammer et al. (2023). Let $p_\theta(\tau) = P_F^\theta(\tau)$ and $\pi(\tau) = R(x)P_B(\tau \mid x)/Z^\star$, with the backward policy, reward, and Z held fixed under differentiation with respect to θ_F. The normalization of the backward policy makes π a probability distribution. Assume common positive support and the regularity needed to differentiate expectations. Write $\delta_\theta = \log(p_\theta/\pi) + c$, where $c = \log Z - \log Z^\star$ is independent of θ_F. The score $s_\theta = \nabla_{\theta_F} \log p_\theta$ has zero mean, since $\mathbb{E}_{p_\theta}[s_\theta] = \nabla_{\theta_F} \sum_\tau p_\theta(\tau) = 0$. Therefore

$$\begin{aligned} \mathbb{E}_{p_\theta}[\nabla_{\theta_F} \delta_\theta^2] &= 2\mathbb{E}_{p_\theta}[\{\log(p_\theta/\pi) + c\} s_\theta] = 2\mathbb{E}_{p_\theta}[\log(p_\theta/\pi) s_\theta], \\ \nabla_{\theta_F} D_{\mathrm{KL}}(p_\theta \| \pi) &= \mathbb{E}_{p_\theta}[\{\log(p_\theta/\pi) + 1\} s_\theta] = \mathbb{E}_{p_\theta}[\log(p_\theta/\pi) s_\theta]. \end{aligned} \tag{20}$$

Combining these equalities proves the factor of two in Equation 7. The identity does not require $Z = Z^\star$: its constant offset cancels by the zero-mean score property. The backward policy may be learned through separate parameters, but shared forward–backward parameters would introduce additional derivative terms.

D.4 PROOF OF PROPOSITION 1

Write the semi-gradient TB and NRD population dynamics as

$$\begin{aligned} \dot\theta_{\mathrm{TB}} &= -2\, \mathbb{E}_{\tau \sim Q_\theta}[\delta_\theta(\tau) \nabla_\theta \delta_\theta(\tau)], \\ \dot\theta_{\mathrm{NRD}} &= \dot\theta_{\mathrm{TB}} + \mathbb{E}_{\tau \sim Q_\theta}\big[L_\theta^-(\tau) \nabla_\theta \log Q_\theta(\tau)\big]. \end{aligned} \tag{21}$$

At a balanced solution $\theta^\star$, $\delta_{\theta^\star}(\tau) = 0$ for every trajectory. Hence both integrands in Equation 21 vanish at $\theta^\star$. Differentiating the TB field at this point, every term arising from the variation of Q_θ or from the Hessian of δ_θ is multiplied by the residual and therefore vanishes. We obtain

$$\left.\frac{\partial \dot\theta_{\mathrm{TB}}}{\partial \theta}\right|_{\theta^\star} = -2\, \mathbb{E}_{\tau \sim Q_{\theta^\star}}\big[\, \nabla_\theta \delta_\theta(\tau) \nabla_\theta \delta_\theta(\tau)^\top \big|_{\theta=\theta^\star} \big]. \tag{22}$$

For the NRD correction, the scalar function $x \mapsto x^2\,\mathbf{1}\{x<0\}$ is continuously differentiable at the origin, with both its value and first derivative equal to zero. Consequently, the derivative of the complete score correction also vanishes at $\theta^\star$. The two Jacobians are therefore equal, which proves the result.

D.5 Proof of Theorem 1

Let

$$q_i = (1-\varepsilon)p_i + \varepsilon\mu_i, \qquad \bar{\delta}_q = \sum_i q_i\delta_i, \qquad \overline{L^-}_p = \sum_i p_i L_i^-. \tag{23}$$

Since the forward distribution is parameterized by free logits Φ,

$$\frac{\partial \log p_i}{\partial \Phi_j} = \mathbf{1}\{i=j\} - p_j. \tag{24}$$

The exact-expectation dynamics of TB and NRD are therefore

$$\begin{aligned} \dot{\log Z} &= -2\bar{\delta}_q, \\ \dot{\Phi}_i^{\mathrm{TB}} &= -2\big(q_i\delta_i - p_i\bar{\delta}_q\big), \\ \dot{\Phi}_i^{\mathrm{NRD}} &= -2\big(q_i\delta_i - p_i\bar{\delta}_q\big) + (1-\varepsilon)p_i\Big(L_i^- - \overline{L^-}_p\Big). \end{aligned} \tag{25}$$

The last term follows from

$$\nabla_\Phi \log q_i = \frac{(1-\varepsilon)p_i}{q_i}\nabla_\Phi \log p_i. \tag{26}$$

If Trajectory Balance holds on every trajectory, then $\delta_i = L_i^- = 0$ for all i, so all three equations in Equation 25 vanish. Every balanced solution is therefore a fixed point of both dynamics. Conversely, consider a fixed point of NRD. The normalizer equation gives $\bar{\delta}_q = 0$. If the residuals were not all zero, the full support of q would imply the existence of an index m with $\delta_m > 0$. For this trajectory $L_m^- = 0$, and Equation 25 gives

$$\dot{\Phi}_m^{\mathrm{NRD}} = -2q_m\delta_m - (1-\varepsilon)p_m\overline{L^-}_p < 0, \tag{27}$$

contradicting stationarity. Hence $\delta_i = 0$ for every i. The same conclusion follows immediately for TB, since at a fixed point $\bar{\delta}_q = 0$ and

$$0 = \dot{\Phi}_i^{\mathrm{TB}} = -2q_i\delta_i. \tag{28}$$

Because $q_i > 0$, every residual must vanish. Finally, $\delta_i = 0$ implies

$$Zp_i = R(x_i)P_B(\tau_i \mid x_i). \tag{29}$$

Summing over all trajectories and using the normalization of $P_B(\cdot \mid x)$ gives

$$Z = \sum_i R(x_i)P_B(\tau_i \mid x_i) = \sum_x R(x) = Z^\star. \tag{30}$$

It follows that $p_i = p_i^\star$ for every i. Thus the two dynamics have the same unique fixed point in (p, Z) coordinates:

$$\mathrm{Fix}(\mathrm{NRD}) = \mathrm{Fix}(\mathrm{TB}) = \{(p^\star, Z^\star)\}. \tag{31}$$

The usual additive constant in the logits leaves p unchanged.

D.6 Spurious fixed points of the full gradient and Naive NRD

For comparison, consider the on-policy two-trajectory dynamics

$$\dot{\Phi}_i = p_i\Big[-2(\delta_i - \bar{\delta}) + s\Big(\delta_i^2 - \overline{\delta^2}\Big)\Big], \qquad \dot{\log Z} = -2\bar{\delta}, \tag{32}$$

where $s=-1$ gives the full expected-loss gradient and $s=+1$ gives Naive NRD. Take

$$p_1 = \frac{1}{4}, \qquad p_2 = \frac{3}{4}. \tag{33}$$

For the full gradient, choose $\delta_1 = -3$ and $\delta_2 = 1$. Then $\bar{\delta} = 0$ and $\overline{\delta^2} = 3$, and

$$\begin{aligned} -2\delta_1 - (\delta_1^2 - \overline{\delta^2}) &= 6 - (9 - 3) = 0, \\ -2\delta_2 - (\delta_2^2 - \overline{\delta^2}) &= -2 - (1 - 3) = 0. \end{aligned} \tag{34}$$

Thus Equation 32 is stationary despite the nonzero residuals. For Naive NRD, choose instead $\delta_1 = 3$ and $\delta_2 = -1$. Again $\bar{\delta} = 0$ and $\overline{\delta^2} = 3$, while

$$\begin{aligned} -2\delta_1 + (\delta_1^2 - \overline{\delta^2}) &= -6 + (9 - 3) = 0, \\ -2\delta_2 + (\delta_2^2 - \overline{\delta^2}) &= 2 + (1 - 3) = 0. \end{aligned} \tag{35}$$

These residual configurations correspond to valid Trajectory Balance problems. Given either pair (p_i, δ_i) above, define

$$C = \sum_i p_i e^{-\delta_i}, \qquad p_i^\star = \frac{p_i e^{-\delta_i}}{C}, \qquad \log \frac{Z}{Z^\star} = -\log C. \tag{36}$$

Then $\delta_i = \log(Z/Z^\star) + \log(p_i/p_i^\star)$ exactly. Taking two terminal objects with $P_B(\tau_i \mid x_i) = 1$ and rewards proportional to $p_i^\star$ realizes these examples as valid GFlowNet instances.

D.7 Proof of Theorem 2

Let

$$r_i = \log \frac{p_i}{p_i^\star}, \tag{37}$$

and let i_+ and i_- attain the maximum and minimum in the definition of D. Since

$$\delta_i = \log \frac{Z}{Z^\star} + r_i, \tag{38}$$

the same indices maximize and minimize the residual. The only difference between the NRD and TB logit fields in Equation 25 is

$$(1 - \varepsilon) p_i \Big(L_i^- - \overline{L^-}_p \Big). \tag{39}$$

The map $x \mapsto x^2 \mathbf{1}\{x < 0\}$ is non-increasing. Hence

$$L_{i_+}^- \le \overline{L^-}_p \le L_{i_-}^- . \tag{40}$$

The correction in Equation 39 is therefore non-positive at a maximizing trajectory and non-negative at a minimizing trajectory. Since

$$\dot{r}_i = \dot{\Phi}_i - \sum_j p_j \dot{\Phi}_j, \tag{41}$$

the common softmax term cancels when differentiating the difference between the maximum and minimum log-ratios. At every differentiability point,

$$\dot{D}_{\mathrm{NRD}} - \dot{D}_{\mathrm{TB}} = (1 - \varepsilon) \Big[p_{i_+} \Big(L_{i_+}^- - \overline{L^-}_p \Big) - p_{i_-} \Big(L_{i_-}^- - \overline{L^-}_p \Big) \Big] \le 0. \tag{42}$$

When either extremum is attained by more than one trajectory, the upper Dini derivative of the maximum is the largest derivative among the active maximizers, while that of the minimum is the smallest derivative among the active minimizers. The same sign argument applies to every active index, which proves the corresponding Dini-derivative inequality.

D.8 Proof of Corollary 1

We first consider the on-policy case. Let $\bar{\delta} = \sum_i p_i \delta_i$, and choose i_+ and i_- attaining the maximum and minimum log-ratios. Since the residuals differ from the log-ratios by the same additive constant,

$$\delta_{i_+} - \delta_{i_-} = D. \tag{43}$$

For TB,

$$\dot{D}_{\mathrm{TB}} = -2 p_{i_+} \big(\delta_{i_+} - \bar{\delta} \big) - 2 p_{i_-} \big(\bar{\delta} - \delta_{i_-} \big) < 0 \tag{44}$$

whenever $D > 0$. Theorem 2 then gives $\dot{D}_{\mathrm{NRD}} \le \dot{D}_{\mathrm{TB}} < 0$. To establish convergence, let $D_0 = D(0)$. Since D is non-increasing, all log-ratios $r_i = \log(p_i/p_i^\star)$ remain inside an interval of width D_0. Writing $m = \min_i r_i$ and using $\sum_i p_i^\star e^{r_i} = 1$ yields

$$-D_0 \le m \le 0. \tag{45}$$

Therefore

$$p_i(t) = p_i^\star e^{r_i(t)} \ge c, \qquad c := e^{-D_0} \min_i p_i^\star > 0. \tag{46}$$

The weighted-average identities then give

$$\begin{aligned} \delta_{i_+} - \bar{\delta} &= \sum_j p_j(\delta_{i_+} - \delta_j) \ge p_{i_-} D \ge cD, \\ \bar{\delta} - \delta_{i_-} &= \sum_j p_j(\delta_j - \delta_{i_-}) \ge p_{i_+} D \ge cD. \end{aligned} \tag{47}$$

Consequently,

$$\dot{D}_{\mathrm{NRD}} \le \dot{D}_{\mathrm{TB}} \le -4c^2 D, \tag{48}$$

and hence

$$D(t) \le D(0) e^{-4c^2 t}. \tag{49}$$

Thus $D(t) \to 0$. Since $\sum_i p_i^\star e^{r_i(t)} = 1$, the common limiting value of the r_i must be zero, and therefore $p(t) \to p^\star$. It remains to consider the normalizer. Writing $\alpha = \log(Z/Z^\star)$,

$$\bar{\delta} = \alpha + \sum_i p_i \log \frac{p_i}{p_i^\star} = \alpha + D_{\mathrm{KL}}(p \| p^\star), \tag{50}$$

and hence

$$\dot{\alpha} = -2(\alpha + D_{\mathrm{KL}}(p \| p^\star)). \tag{51}$$

Since $p(t) \to p^\star$, the KL term converges to zero, and the stable linear equation above gives $\alpha(t) \to 0$. Therefore $Z(t) \to Z^\star$. We finally prove the strict comparison under conditional normalization. Suppose

$$\bar{\delta}_q = \mathbb{E}_{Q_\varepsilon}[\delta] = 0 \tag{52}$$

and $D > 0$. Since q has full support, the largest and smallest residuals must satisfy

$$\delta_{i_+} > 0 > \delta_{i_-}. \tag{53}$$

The TB part of Equation 25 therefore gives

$$\dot{D}_{\mathrm{TB}} = -2\big(q_{i_+}\delta_{i_+} - q_{i_-}\delta_{i_-}\big) < 0. \tag{54}$$

Moreover, $L^-_{i_+} = 0$ while $\overline{L^-}_p > 0$. For every $\varepsilon < 1$, the NRD correction at the maximizing trajectory is therefore strictly negative, while its correction at the minimizing trajectory is non-negative. Hence

$$\dot{D}_{\mathrm{NRD}} < \dot{D}_{\mathrm{TB}} < 0. \tag{55}$$

For $\varepsilon = 1$, $Q_\varepsilon = \mu$ is independent of the forward policy and the NRD score term vanishes.

D.9 Proof of Theorem 3

For the exponential family in Theorem 3,

$$\log \frac{p_\eta(\tau)}{p_{\eta^\star}(\tau)} = (\eta - \eta^\star)T(\tau) - (A(\eta) - A(\eta^\star)). \tag{56}$$

Under fast normalization, $\mathbb{E}_{p_\eta}[\delta_\eta] = 0$. Subtracting the expectation of the log-ratio therefore gives

$$\delta_\eta(\tau) = (\eta - \eta^\star)\big(T(\tau) - \mathbb{E}_{p_\eta}[T]\big) = (\eta - \eta^\star)\partial_\eta \log p_\eta(\tau). \tag{57}$$

The TB and NRD parameter dynamics are

$$\begin{aligned} \dot{\eta}_{\mathrm{TB}} &= -2\mathbb{E}_{p_\eta}[\delta_\eta \partial_\eta \log p_\eta], \\ \dot{\eta}_{\mathrm{NRD}} &= \dot{\eta}_{\mathrm{TB}} + \mathbb{E}_{p_\eta}\big[\delta_\eta^2 \mathbf{1}\{\delta_\eta < 0\}\partial_\eta \log p_\eta\big]. \end{aligned} \tag{58}$$

Multiplying by $\eta - \eta^\star$ and using Equation 57 yields

$$\begin{aligned}(\eta - \eta^\star)\dot{\eta}_{\mathrm{TB}} &= -2\mathbb{E}_{p_\eta}\big[\delta_\eta^2\big],\\ (\eta - \eta^\star)\dot{\eta}_{\mathrm{NRD}} &= -2\mathbb{E}_{p_\eta}\big[\delta_\eta^2\big] + \mathbb{E}_{p_\eta}\big[\delta_\eta^3 \mathbf{1}\{\delta_\eta < 0\}\big].\end{aligned} \tag{59}$$

For $\eta \neq \eta^\star$, the sufficient statistic is nonconstant and the centered random variable in Equation 57 takes both positive and negative values. Hence the last expectation in Equation 59 is strictly negative. This proves

$$(\eta - \eta^\star)\dot{\eta}_{\mathrm{NRD}} < (\eta - \eta^\star)\dot{\eta}_{\mathrm{TB}} < 0. \tag{60}$$

Both dynamics therefore move monotonically toward $\eta^\star$. A non-target limiting value would contradict the strict inequality above by continuity, so both converge to $\eta^\star$. For a regular one-dimensional exponential family, the Jeffreys divergence is

$$J(p_\eta, p_{\eta^\star}) = (\eta - \eta^\star)\big(A'(\eta) - A'(\eta^\star)\big). \tag{61}$$

Indeed, the log-partition terms cancel when the two KL divergences are added, and $A'(\eta) = \mathbb{E}_{p_\eta}[T]$. The divergence is finite for interior natural parameters, including families with unbounded sufficient statistics. Moreover,

$$\partial_\eta J = A'(\eta) - A'(\eta^\star) + (\eta - \eta^\star)A''(\eta). \tag{62}$$

Since $A''(\eta) = \mathrm{Var}_{p_\eta}(T) > 0$, this derivative has the sign of $\eta - \eta^\star$. Thus J strictly decreases as the parameter moves toward $\eta^\star$, and vanishes at the target. Suppose first that $\eta_0 > \eta^\star$. At every common $\eta > \eta^\star$,

$$\dot{\eta}_{\mathrm{NRD}} < \dot{\eta}_{\mathrm{TB}} < 0, \tag{63}$$

so NRD has strictly greater speed toward the target. If $\eta_0 < \eta^\star$, division by the negative quantity $\eta - \eta^\star$ reverses the inequalities and gives $\dot{\eta}_{\mathrm{NRD}} > \dot{\eta}_{\mathrm{TB}} > 0$, with the same interpretation. Since these are scalar autonomous dynamics, the time required to cross any interval between the initialization and the target is the integral of the reciprocal speed over that interval. The strict pointwise speed comparison therefore yields

$$T_{\mathrm{NRD}}(\epsilon) < T_{\mathrm{TB}}(\epsilon) \tag{64}$$

for every $0 < \epsilon < J(p_{\eta_0}, p_{\eta^\star})$: each such threshold corresponds to a unique parameter between the initialization and the target. The standard scalar comparison theorem also gives $J_{\mathrm{NRD}}(t) < J_{\mathrm{TB}}(t)$ for every $t > 0$.

D.10 Two-trajectory specialization

Assume $p > p^\star$; the other case follows after exchanging the trajectory labels. The discrepancy is

$$D = \log \frac{p/(1-p)}{p^\star/(1-p^\star)}. \tag{65}$$

Under fast normalization, the two residuals and the average negative-residual loss are

$$\delta_1 = (1-p)D, \qquad \delta_2 = -pD, \qquad \overline{L^-} = (1-p)p^2D^2. \tag{66}$$

For TB, the two free logits therefore satisfy

$$\dot{\Phi}_1^{\mathrm{TB}} = -2p(1-p)D, \qquad \dot{\Phi}_2^{\mathrm{TB}} = 2p(1-p)D. \tag{67}$$

Since $D = \Phi_1 - \Phi_2 - \mathrm{const}$,

$$\dot{D}_{\mathrm{TB}} = -4p(1-p)D. \tag{68}$$

Only the second trajectory has a negative residual. The NRD score correction to the two logits is

$$-p^3(1-p)D^2 \qquad \text{and} \qquad p^3(1-p)D^2, \tag{69}$$

respectively. Subtracting the two logit dynamics gives

$$\dot{D}_{\mathrm{NRD}} = -4p(1-p)D - 2p^3(1-p)D^2. \tag{70}$$

These equations use two free logits; relative to a single Bernoulli natural parameter, both fields share the same factor-of-two time rescaling. The scalar speed comparison therefore gives the trajectory and hitting-time ordering directly. Here $J = (p - p^\star)D$ is strictly increasing in $D > 0$, so a positive Jeffreys threshold is equivalent to a positive threshold on D along this branch, consistently with

Theorem 3. It remains to establish the unbounded additive hitting-time advantage. Along the branch $p > p^\star$,

$$\frac{p(D)}{1-p(D)} = \frac{p^\star}{1-p^\star} e^D. \tag{71}$$

For a fixed Jeffreys threshold $\epsilon > 0$, let $d_\epsilon > 0$ be the unique solution of $(p(d_\epsilon) - p^\star)d_\epsilon = \epsilon$. The corresponding hitting times, for $D(0) > d_\epsilon$, are

$$\begin{aligned} T_{\mathrm{TB}}(\epsilon) &= \int_{d_\epsilon}^{D(0)} \frac{dD}{4p(D)(1-p(D))D}, \\ T_{\mathrm{NRD}}(\epsilon) &= \int_{d_\epsilon}^{D(0)} \frac{dD}{4p(D)(1-p(D))D\big(1+\frac{1}{2}p(D)^2 D\big)}. \end{aligned} \tag{72}$$

Their difference is

$$T_{\mathrm{TB}}(\epsilon) - T_{\mathrm{NRD}}(\epsilon) = \int_{d_\epsilon}^{D(0)} \frac{p(D)}{8(1-p(D))\big(1+\frac{1}{2}p(D)^2 D\big)}\, dD. \tag{73}$$

Since

$$\frac{p(D)}{1-p(D)} = \frac{p^\star}{1-p^\star} e^D, \tag{74}$$

the integrand grows asymptotically as a positive constant times e^D/D. Its integral therefore diverges as $D(0) \to \infty$, proving

$$T_{\mathrm{TB}}(\epsilon) - T_{\mathrm{NRD}}(\epsilon) \longrightarrow \infty. \tag{75}$$